REMARKS

Claim 19 has been amended. No claims have been added or cancelled. Therefore claims 1-88 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 112, Second Paragraph, Rejection:

The Examiner rejected claims 7, 14, 20, 30, 42, 48, 59, 71, 76 and 83 under 35 U.S.C. § 112, second paragraph, as being indefinite. Specifically, regarding claims 7, 20, 42 and 76, the Examiner submits that it is unclear what is meant by "discover the one or more high-level functions" by accessing information stored in a metadata repository. The Examiner asserts, "Since Applicants' specification provides no specific definition, it is construed that the discovering process is equivalent to the nominal indexing, searching, or keying over a metadata array of file."

Applicants submit that the Examiner has misconstrued claims 7, 20, 42, and 76. These claims do not recite "discover the one or more high-level functions by accessing information stored in a metadata repository," as the Examiner suggests. Instead, they recite, for example, "accessing a metadata repository comprising information describing one or more high-level functions of the external system to discover the one or more high-level functions provided by the external system," and similar limitations. The plain language of these claims would be clear to anyone of ordinary skill in the art at the time the invention was made. Furthermore, it is not clear what the Examiner means by "it is construed that the discovering process is equivalent to the nominal indexing, searching, or keying over a metadata array of file." Applicants' claims do not recite anything about indexing, searching, or keying, and the phrase "over a metadata array of file" is meaningless.

Regarding claims 14, 30, 48, 59, 71, and 83, the Examiner submits that it is unclear what constitutes the so-called "J2EE Connector Architecture." Specifically, the

Examiner notes that Applicants' specification states, "J2EE CA specifies a required contract **between** the connector and the container for managing resource pooling, transactions and security" (emphasis Examiner's.) The Examiner submits that based on this statement it appears that the connector is outside the container. Applicants respectfully disagree. A contract between two hardware or software components in a computing system is not a physical structure or logical component **between** the components, nor does it imply anything about the physical or logical location of the components with respect to each other, as the Examiner suggests. Instead, such a contract is set of requirements for hardware or software resources, and/or for implementation of protocols, etc., that must be met by the two components in order for them to be compliant with a specification. The fact that a specific contract is required between the connector and the container has nothing to do with whether the connector is inside or outside the container, as the Examiner suggests. As would be understood by anyone of ordinary skill in the art, such contracts may exist between components of a computing system having any one of various relationships, including a hierarchical relationship in which the connector is comprised within the container.

The Examiner further submits that based on the statement, "J2EE CA also specifies a Common Client Interface (CCI) for client interaction with the connector, which the connector may choose to implement," it appears that CCI is an option to the architecture because the connector "may choose" to implement it. The Examiner submits that because the CCI is optional, this would render the claims indefinite if the Applicants' paragraph including these sentences is read into the claims. Therefore, for prior art rejection the "J2EE CA" connector is construed as a generic connector for connecting between a Java based platform to any enterprise system. The Examiner requests clarification in response. Applicants strongly disagree that these claims are indefinite. Applicants assert that the limitation "wherein the connector is a Java 2 Enterprise Edition Connector Architecture (J2EE CA) connector" clearly refers to a connector that is compliant with the J2EE CA specification. Applicants further assert that a connector may be compliant with the J2EE CA specification with or without implementing the CCI specified by the J2EE CA specification, or any other optional features. Applicants assert,

therefore, that the limitations of these claims neither require nor exclude implementation of the CCI specified by the J2EE CA specification, and are not made unclear by the omission of a reference thereto. "Breadth of a claim is not to be equated with indefiniteness." M.P.E.P. § 2173.04; *In re Miller*, 441 F.2d 689, 169 USPQ 597 (CCPA 1971).

Furthermore, as stated in MPEP 2143.03, a claim limitation which is considered indefinite cannot be disregarded. Therefore, it is improper for the Examiner to construe the connector of claims 14, 30, 48, 59, 71, and 83 as "a generic connector for connecting between a Java based platform to any enterprise system," rather than as "a Java 2 Enterprise Edition Connector Architecture (J2EE CA) connector," as recited in these claims.

For at least the reasons above, Applicants respectfully request removal of the rejection of claims 7, 14, 20, 30, 42, 48, 59, 71, 76 and 83 under 35 U.S.C. § 112, second paragraph.

Section 102(e) Rejection:

The Examiner rejected claims 1, 2, 9-19, 24, 26-38, 43, 45 and 48-50 under 35 U.S.C. § 102(e) as being anticipated by Sheriff et al. (U.S. Patent 6,854,122) (hereinafter "Sheriff"). Applicants respectfully traverse this rejection for at least the following reasons.

Regarding claim 1, contrary to the Examiner's assertion, Sheriff fails to teach or suggest *the container accessing metadata corresponding to the high-level function, wherein the metadata describes the high-level function of the external system; the container performing one or more transformations on the data object specifying the high-level function in accordance with the metadata to produce a data object including information for driving a connector to the external system to make a plurality of low-level calls to the external system to perform the high-level function.* The Examiner cites

column 3, line 50 – column 7, line 10 as teaching the first limitation above ("the container accessing metadata..."). However, this portion of Sheriff does not disclose metadata at all, much less metadata that describes the high-level function of the external system. Instead, this portion of Sheriff includes a textual description (similar to a software specification) of various methods supported by the CIM client adapter 14 and various CIM operations supported by the CIM/WMI mapper. For example, column 6, line 46 – column 7, line 10 does not constitute metadata used by a mapper, as suggested by the Examiner. Instead, this passage contains descriptions of supported functions, such as would be found in a software specification for one embodiment of Sheriff's mapper. There is nothing in Sheriff that teaches or suggests that these descriptions themselves have any function in the system of Sheriff. They are merely included in Sheriff's specification as examples of the CIM functions supported by a mapper in one embodiment of Sheriff's system. Therefore, the Examiner's interpretation of this portion of Sheriff, and the Examiner's remarks regarding it, are clearly incorrect. There is nothing in Sheriff that teaches or suggests a container accessing these descriptions or any metadata corresponding to the methods described therein. Therefore, these citations clearly do not have anything to do with the container accessing metadata corresponding to the high-level function, wherein the metadata describes the high-level function of the external system, as recited in claim 1.

The Examiner cites column 3, lines 25-38 and 45-49, and column 7, line 12-34 as teaching the second limitation above ("the container performing one or more transformations... in accordance with the metadata...") and submits that this is analogous to "APIs or the routines translated into bytecode." However, none of these citations, or anything else in Sheriff teaches or suggests that translating APIs or other routines into bytecode involves performing one or more transformations on a data object specifying a high-level function in accordance with the metadata, since no such metadata is disclosed in Sheriff.

The Examiner submits that "by treating each of the functions listed at column 3, lines 50-61 as low-level functions, it is clear that a high level function may be specified

to include at least open() and close() because those are commonly needed routines in a high-level application." Applicants assert, however, that while a high-level function may include one or more of these functions, nothing in Sheriff discloses *an application sending to a container a data object specifying a high-level function provided by a system external to the application, the container receiving the data object specifying the high-level function,* and then the container performing the transformations discussed above. Instead, using the Examiner's interpretation, Sheriff discloses CIM/WMI mapper 17 mapping the low-level APIs from Java console 11 (the functions of column 3, lines 50-61) to corresponding low-level WMI functions, as described in columns 6 and 7.

The Examiner further submits, "On the other hand, when treating the functions listed at column 3, lines 50-61 as high-level functions, then each (Java) routine is converted into a set of low-level instructions written in terms of byte code causing the external system to execute the corresponding low-level functions." Applicants assert, however, that there is nothing in Sheriff that teaches or suggests a container receiving these functions and the container transforming them into a set of low-level instructions (byte-code) causing the external system to execute the corresponding low-level function, as the Examiner suggests. Instead, Sheriff describes that the functions received at client adapter 14 originate at Java WPIC 11 and that client adapter 14 implements a set of APIs used by the WPIC 11 to perform operations such as adding, modifying, or deleting a CIM class, CIM instance, and CIM qualifier type in namespace. Furthermore, Sheriff describes the use of the Java Native Interface (JNI) and Java Virtual Machine in column 7, lines 19-32:

> In the present invention, the JNI is used to make the library of the managed server 20 accessible to Java code. The JNI allows code that runs within a Java Virtual Machine (VM) to operate with applications and libraries written in other languages, such as C, and C++ and allows the JNI to be embedded into native applications. The Java Virtual machine is responsible for interpreting Java byte code, and translating this into actions or operating system calls.
>
> With reference to FIG. 2, an application 100 typical of the invention is depicted. A JNI 101 is implemented in conjunction with a VM 102 to serve as a translator between Exceptions 103, and classes 104 on the Java side and Functions 105 and Libraries 106 written in C, on the managed server side.

FIG. 2 of Sheriff depicts that JNI 101 and VM 102 are embedded into application 100 to serve as a translator between Java side Exceptions and Classes and the Functions and Libraries written in C on the managed server side. Therefore, Sheriff does not describe a container performing the transformations recited in claim 1 (in accordance with the accessed metadata), but instead describes an application receiving a CIM communication from a Java console 11, converting it to a WMI communication (without any reference to metadata), and communicating it to a managed server.

Applicants remind the Examiner that anticipation requires the presence in a single prior art reference disclosure of each and every limitation of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The **identical** invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As discussed above, Sheriff clearly fails to disclose all the limitations of Applicants' claim 1. Therefore, Sheriff cannot be said to anticipate claim 1.

For at least the reasons above, the rejection of claim 1 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 18 and 36 were rejected under the same rationale as claim 1 and recite similar limitations regarding a container receiving a high-level function call from an application and the container mapping the high-level function call to low-level functions of an external system, which executes the low-level functions to perform the high-level function. Therefore, the arguments presented above apply with equal force to these claims, as well.

Regarding claim 10, contrary to the Examiner's assertion, Sheriff fails to teach or suggest *wherein said mapping the high-level function to a plurality of low-level function calls to the external system comprises the container accessing metadata corresponding to the high-level function, wherein the metadata maps the high-level function to the plurality*

of low-level function calls to the external system specific to the connector. The Examiner cites FIGs. 1-3 as teaching this limitation and submits that the connector is a CIM/WMI mapper, "which is specific for mapping between Java based routines and window based management data." However, as discussed above regarding claim 1, there is nothing in these figures or elsewhere in Sheriff that teaches or suggests the container accessing metadata when mapping the high-level function to a plurality of low-level function calls. In fact, Sheriff does not teach or suggest the use of metadata at all. Therefore, Sheriff cannot be said to anticipate claim 10.

Therefore, for at least the reasons above, the rejection of claim 10 is unsupported by the cited art and removal thereof is respectfully requested.

Similarly, regarding claim 11, Sheriff fails to teach or suggest *modifying the metadata corresponding to the high-level function to map the high-level function to a plurality of low-level function calls to the external system specific to a different connector to the external system.* The Examiner submits that this is taught by element 17 of FIG. 1, which is different from element 308 of FIG. 3, "which can be obtained by simply modifying the metadata contained in the mapper." However, there is nothing in Sheriff that teaches or suggests "metadata contained in the mapper," much less modifying such metadata, as the Examiner suggests. As discussed above, the use of metadata in mappers is not taught or suggested by Sheriff. In addition, FIG. 1 element 17 is a CIM/WMI mapper for mapping to managed server 20 and FIG. 3 element 308 is a CIM/WBEM mapper for mapping to a different managed server, 311. There is nothing in Sheriff that teaches or suggests that one of these mappers was generated by modifying the other. Furthermore, nothing in Sheriff teaches or suggests a method comprising modifying metadata to map a high-level function to a plurality of low-level function calls specific to a different connector to the (same) external system, as in claim 11. Therefore, Sheriff cannot be said to anticipate claim 11.

For at least the reasons above, the rejection of claim 11 is unsupported by the cited art and removal thereof is respectfully requested.

Regarding claim 19, Sheriff fails to teach or suggest wherein the metadata is accessed from a metadata repository comprising metadata describing a plurality of high-level functions of the external system. The Examiner rejected claim 19 for the same reasons set forth in the rejection of claims 1-2 and 9-17. However, none of these claims includes this limitation. **Since the Examiner failed to address the differences between claim 19 and claims 1-2 and 9-17, the rejection of claim 19 is improper.** Furthermore, since Sheriff fails to teach or suggest the use of metadata, Sheriff also clearly fails to teach or suggest accessing metadata from a metadata repository, as recited in claim 19.

For at least the reasons above, the rejection of claim 19 is unsupported by the cited art and removal thereof is respectfully requested.

Similarly, claims 27-29 and 37-38 recite limitations involving the use of metadata and/or a metadata repository. Therefore, for at least the reasons presented above regarding claims 1, 10, 11, and 19, the rejection of these claims is also unsupported by the cited art and removal thereof is respectfully requested.

Section 103(a) Rejection:

The Examiner rejected claims 3-8, 20-23, 25, 39-42, 44, 46, 47 and 51-88 under 35 U.S.C. § 103(a) as being unpatentable over Sheriff, as applied to claims 1, 2, 9-19, 24, 26-38, 43, 45 and 48-50 above. Applicants respectfully traverse this rejection for at least the following reasons.

The Examiner rejected claims 20-23, 25, 39-42, 44, 46-47, and 51-88 for the same reasons set forth in the rejection of claims 1-19, 24, 26-38, and 48-50 because features of these claims can also be found in claims 1-19, 24, 26-38, and 48-50. However, the scope of many of these claims differs from those of claims 1-19, 24, 26-38, and 48-50. **Since the Examiner failed to specifically address the differences between these claims, the**

Examiner has failed to state a *prima facie* rejection of claims 20-23, 25, 39-42, 44, 46-47, and 51-88.

Further regarding independent claim 51, contrary to the Examiner's assertion, Sheriff fails to teach or suggest *a container, wherein the EIS is external to the container, and wherein the container comprises... means for receiving a high-level function call from the application; means for mapping the high-level function call to a series of low-level function calls to the EIS; and means for driving the connector to make the series of low-level function calls to the EIS.* As discussed above, Sheriff does not teach or suggest a container comprising means to receive a high-level function call from an application and to map it to low-level function calls to an external system, such as the EIS of claim 51. Instead, Sheriff describes a low-level CIM/WMI mapper 17 and an application containing an embedded JNI 101 and VM 102 that serve to translate Java structures to functions and libraries on the managed server side.

Similarly, regarding independent claim 62, Sheriff fails to teach or suggest *a metadata-aware adapter for a connector to an Enterprise Information System (EIS), configured to receive a high-level function call for a high-level functions of the EIS,* to *map the high-level function call to a series of low-level function calls to the EIS* and to *drive the connector to make the series of low-level function calls to the EIS.* Instead, Sheriff describes a low-level CIM/WMI mapper 17 and an application containing an embedded JNI 101 and VM 102 that serve to translate Java structures to functions and libraries on the managed server side, neither of which are described as being metadata-aware adapters for a connector to an Enterprise Information System.

Finally, regarding independent claim 73, Sheriff fails to teach or suggest program instructions computer-executable to *implement an application sending to a container a high-level function call corresponding to a high-level function of a system external to the application; the container mapping the high-level function call to a series of low-level function calls to the external system;* and *the container causing a connector to the external system comprised in the container to make the series of low-level function calls*

to the external system. As discussed above, Sheriff does not teach or suggest an application sending a high-level function call to a container and the container mapping it to low-level function calls to an external system and causing a connector in the container to make the low-level function calls. Instead, Sheriff describes a low-level CIM/WMI mapper 17 and an application containing an embedded JNI 101 and VM 102 that serve to translate Java structures to functions and libraries on the managed server side.

For at least the reasons above the rejection of independent claims 51, 62, and 73 is unsupported by the cited art and removal thereof is respectfully requested.

Regarding claims 3-5, the Examiner admits that Sheriff is silent about how the metadata is stored, but submits that in order to use each of the supported methods listed at column 3, line 64 – column 6, line 29, it is obvious that the metadata should be stored somewhere in a memory space, which can be in a persistent store or a repository for further references. Applicants assert, however, that as discussed above regarding claims 1, 10, 11, and 19, Sheriff fails to teach or suggest the use of metadata in the mapping operation at all. Furthermore, there is nothing in Sheriff that teaches or suggests that metadata is used to support any of the methods listed at column 3, line 64 – column 6, line 29, as the Examiner suggests. Therefore, contrary to the Examiner's assertion, it would not be obvious to store metadata anywhere, much less in a persistent store or repository, as recited in Applicants' claims.

The Examiner further submits that according to the example of FIGs. 1 and 3, the management application is sent from a remote java console 11 external to the container storing the metadata, and that as such, it is clear that there is no need to repopulate the metadata repository in the situation when the application is moved from an unmanaged environment to a managed environment because the container is independent of the application's platform. Again, Applicant asserts that Sheriff does not teach or suggest using metadata or storing metadata in a repository at all. Therefore, the Examiner's comments regarding this limitation of claim 5 do not apply.

For at least the reasons above, the rejection of claims 3-5 is unsupported by the cited art and removal thereof is respectfully requested.

Regarding claim 6, the Examiner submits, "Based on the example of Java-C code translation (FIG. 2), it is an obvious option to also implement the metadata repository in a Java Naming and Directory Interface (JNDI) namespace because the latter is needed for implementing the Java virtual machine." Applicants assert, however, that the use of a metadata repository is not taught or suggested by Sheriff at all, and is not needed for implementing the translation illustrated in FIG. 2. Therefore, contrary to the Examiner's assertion, it would not be obvious to implement such a repository in a JNDI namespace.

For at least the reasons above, the rejection of claim 6 is unsupported by the cited art and removal thereof is respectfully requested.

Similarly, regarding claims 7-8, the Examiner admits that Sheriff does not specifically teach how each of the high-level functions of the external system is discovered in the metadata repository and how each function's associated parameters are retrieved from the repository. The Examiner submits that since each of the supported CIM methods has a corresponding name in the CIM/WMI mapper with the original CIM name embedded in it (e.g., column 6, line 46 – column 7, line 10), it is obvious to one of ordinary skill in the art that the CIM methods of the external system (i.e., the CIM methods) can be found from the list of names (which constitute part of the metadata) in the mapper by using a hashing technique because hashing is a popular technique for effectively organizing string arrays. Applicants again assert, however, that column 6, line 46 – column 7, line 10 are not lines of code comprised in the mapper of Sheriff, nor do they constitute metadata used by a mapper, as suggested by the Examiner. Instead, they are merely descriptions of supported functions, such as would be found in a software specification for one embodiment of Sheriff's system. There is nothing in Sheriff that teaches or suggests that these descriptions themselves have any function in the system of Sheriff. They are merely included in Sheriff's specification as examples of the CIM

functions supported in one embodiment. Therefore, the Examiner's interpretation of this portion of Sheriff, and his remarks regarding it, are incorrect.

For at least the reasons above, the rejection of claims 7-8 is unsupported by the cited art and removal thereof is respectfully requested.

Similarly, claims 20-23, 25, 39-42, 44, 46-47, 52-56, 63-65, 69-70, 75-78, and 80-82 recite limitations involving the use of metadata and/or a metadata repository. Therefore, for at least the reasons presented above regarding claims 1, 3-8, 10, 11, and 19, the rejection of these claims is also unsupported by the cited art and removal thereof is respectfully requested.

Applicants also assert that numerous other ones of the dependent claims recite further distinctions over the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

Applicants submit the application is in condition for allowance, and prompt notice to that effect is respectfully requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above-referenced application from becoming abandoned, Applicants hereby petition for such an extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-92801/RCK.

Also enclosed herewith are the following items:

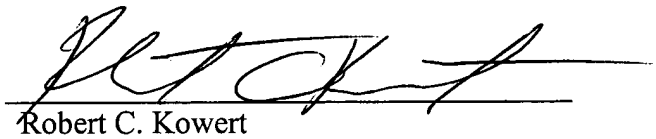
☒ Return Receipt Postcard

☐ Petition for Extension of Time

☐ Notice of Change of Address

☐ Other:

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: June 29, 2006